Claims

WHAT IS CLAIMED IS:

5   1.   A method of providing concurrent access to data on cross-computing
environments, comprising:

receiving a first request for the data from a first computing environment;

receiving a second request for the data from a second computing

environment, wherein the second computing environment is different from the first

10   computing environment;

identifying a first message file to service the requests; and

concurrently providing access to the first message file in order to load the

data as directed by the first request from the first computing environment and the

second request from the second computing environment.

15

2.   The method of claim 1, further comprising loading the first message file into
a memory before providing the message file to the requests.

3.   The method of claim 1, further comprising loading the first message file

20   into a data store before providing the message file to the requests.

4.   The method of claim 1, wherein in identifying the first message file, the first
message file is identified with a first language associated with the requests.

25   5.   The method of claim 1, wherein in providing the first message file, the first
message file is represented in a generic file format.

30

6. The method of claim 1, further comprising:

loading the first message file into a memory;

receiving a third request from a third computing environment for the data, wherein the third request requires a second message file associated with a different language than required by the first request and the second request;

identifying the second message file;

loading the second message file into the memory concurrently with the loaded first message file; and

concurrently providing access to the second message file in order to load the data as required by the third request from the third computing environment while access is provided to the first message file to satisfy the first request and the second request.

7. The method of claim 6, wherein in receiving the third request, the third computing environment is the same as the first computing environment or the second computing environment.

8. The method of claim 1, wherein in receiving the third request, the third computing environment is the different from the first computing environment and the second computing environment.

9. A method of providing access to data in cross-languages and on cross-computing environments, comprising:

receiving a first request for the data, wherein the first request requires the data to be provided in a first language on a first computing environment;

receiving a second request for the data, wherein the second request requires the data to be provided in a second language on a second computing environment;

identifying a first message file of the data for the first language;

identifying a second message file of the data for the second language; and

concurrently providing the first message file and the second message file to service the first request and the second request in the first computing environment

and the second computing environment.

10. The method of claim 9, further comprising concurrently loading the first message file and the second message file into a data structure.

5

11. The method of claim 10, wherein in loading the first message file and the second message file, the data structure is a table to provide the data in the first language and the second language, and wherein the data structure is loaded to a volatile memory.

10

12. The method of claim 10, wherein in loading the first message file and the second message file, the data structure is an indexed set of records to provide the data in the first language and the second language, and wherein the data structure is loaded to a data store.

15

13. The method of claim 12, wherein in loading the first message file and the second message file, the data structure is loaded to fields of the data store and the data store is a database.

20 14. The method of claim 13, wherein in loading the first message file and the second message file, the database is a flexible and adaptable information manager database.

15. The method of claim 9, wherein in identifying the message files, the

25 message files are in an - extensible markup language file format.

30

16.   A system to provide access to data in cross-language formats and on cross computing environments, comprising:

a first request for the data from a first computing environment;

a second request for the data from a second computing environment;

5   a first message file;

a second message file; and

a set of language manager executable instructions operable to identify the first message file and load it into a first data structure representing the data in a first language, the set of language manager executable instructions also is further

10   operable to identify the second message file and load it into a second data structure representing the data in a second language.

17.   The system of claim 16, wherein the first computing environment is different from the second computing environment and the first language is different from the

15   second language.

18.   The system of claim 16, wherein the set of language manager executable instructions loads the first message file and the second message file into a memory.

20   19.   The system of claim 16, wherein the set of language manager executable instructions loads the first message file and the second message file into a data store.

20.   The system of claim 16, wherein at least one of the requests is received over the Internet using a set of browser executable instructions.

25

21.   The system of claim 16, wherein the set of language manager executable instructions retrieves the data from the first data structure and the second data structure to satisfy the first request and the second request.

30

22.   The system of claim 16, further comprising a single application programming interface library providing an interface between the set of language manager executable instructions and the requests from the computing environments.

5   23.   The system of claim 16, wherein the requests and the message files are represented in an extensible markup language format.

24.   A cross-computing environment message file, comprising:

a message identifier associated with language data used to represent data in a

10 requested language; and

tagging data operable to identify instructions associated with providing the data in the requested language to a first computing environment and to a second computing environment, wherein the first computing environment is different from the second computing environment.

15

25.   The message file of claim 24, wherein the message file interfaces with a first request originating from the first computing environment and a second request originating from the second computing environment, wherein the requests are made to acquire the data in the requested language.

20

26.   The message file of claim 24, wherein messages from the message file are transferred over a network.

27.   The message file of claim 24, wherein the instructions generate a table

25 having the requested data in the requested language.

28.   The message file of claim 24, wherein the tagged data is represented in an extensible markup language format.

30 29.   The message file of claim 24, wherein the instructions load the data as a table in at least one of a memory and a data store.